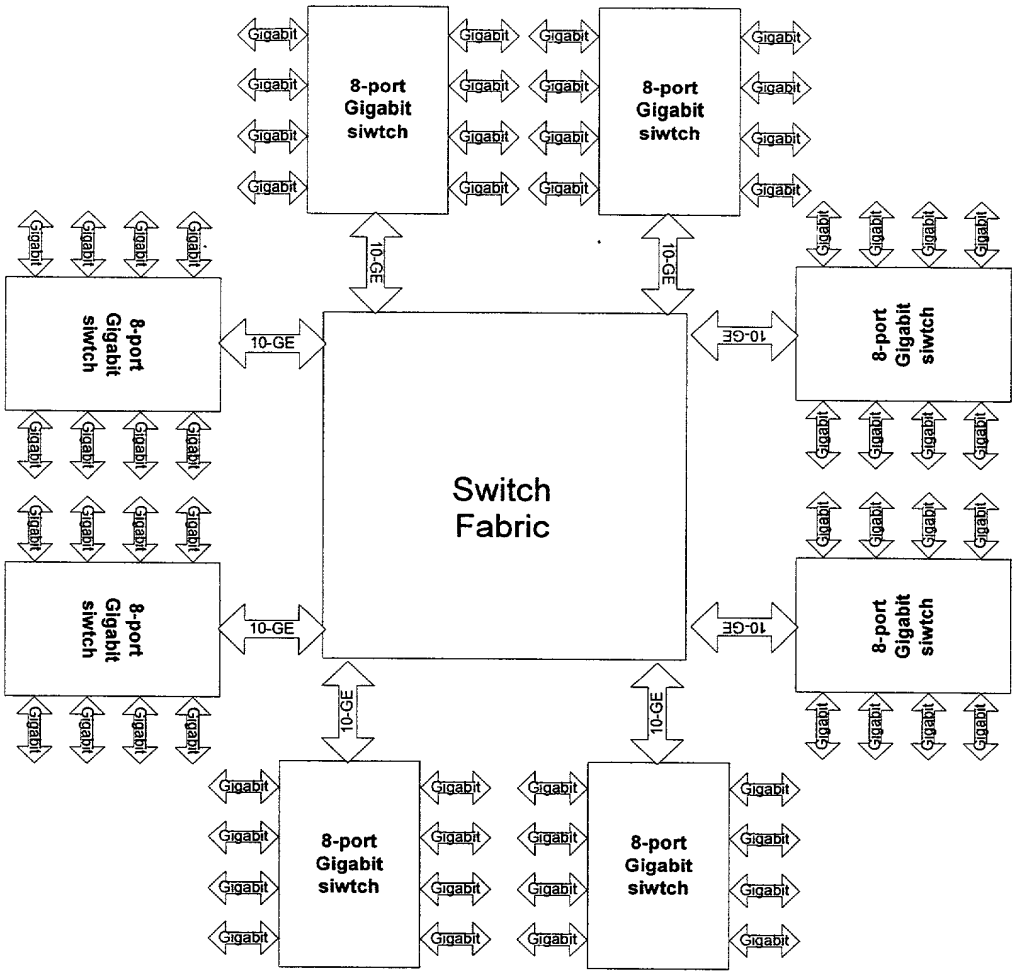
8-port Gigabit siwtch
8-port Gigabit siwtch
8-port Gigabit siwtch
8-port Gigabit siwtch
8-port Gigabit siwtch
8-port Gigabit siwtch
Switch Fabric
10-GE
Gigabit

Fig. 1

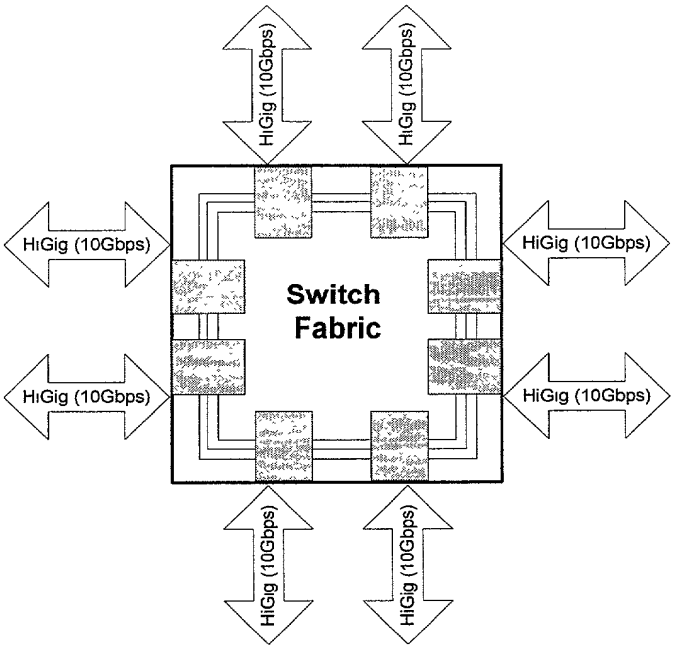
HiGig (10Gbps)
HiGig (10Gbps)
HiGig (10Gbps)
HiGig (10Gbps)
Switch Fabric
HiGig (10Gbps)
HiGig (10Gbps)
HiGig (10Gbps)
HiGig (10Gbps)

Fig. 2

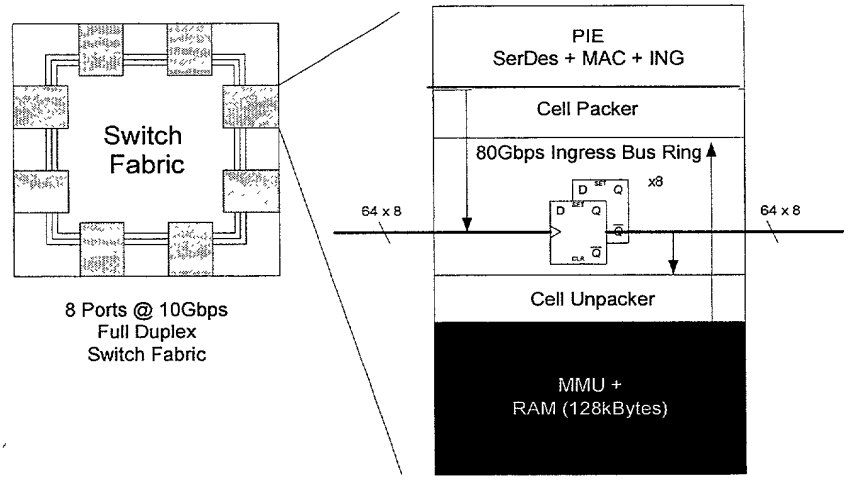
Switch Fabric
8 Ports @ 10Gbps
Full Duplex
Switch Fabric
PIE
SerDes + MAC + ING
Cell Packer
80Gbps Ingress Bus Ring
x8
64 x 8
64 x 8
Cell Unpacker
MMU +
RAM (128kBytes)

Fig. 3

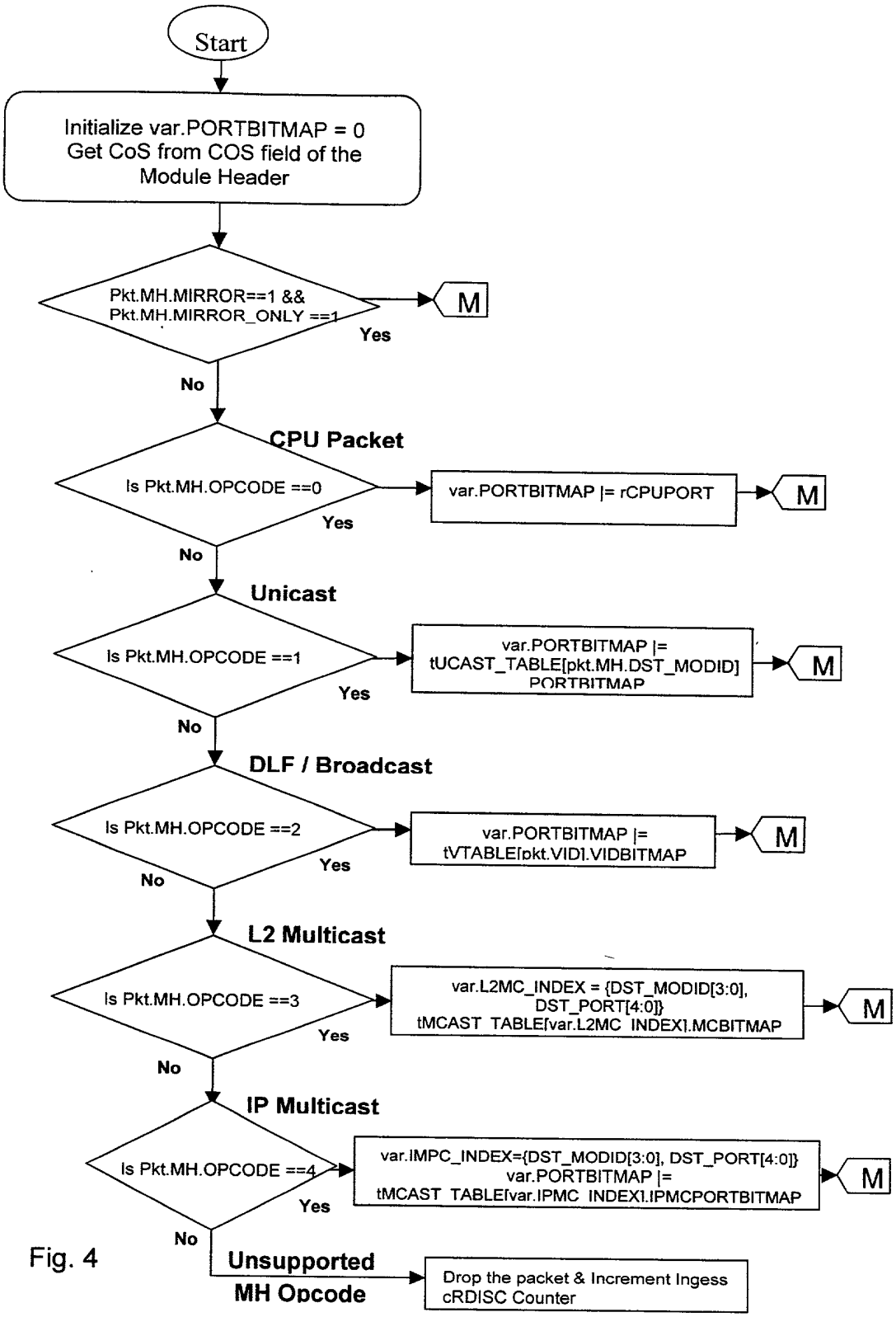
Start
Initialize var.PORTBITMAP = 0
Get CoS from COS field of the Module Header
Pkt.MH.MIRROR==1 && Pkt.MH.MIRROR_ONLY ==1
Yes
M
No
CPU Packet
Is Pkt.MH.OPCODE ==0
Yes
var.PORTBITMAP |= rCPUPORT
M
No
Unicast
Is Pkt.MH.OPCODE ==1
Yes
var.PORTBITMAP |= tUCAST_TABLE[pkt.MH.DST_MODID] PORTBITMAP
M
No
DLF / Broadcast
Is Pkt.MH.OPCODE ==2
Yes
var.PORTBITMAP |= tVTABLE[pkt.VID].VIDBITMAP
M
No
L2 Multicast
Is Pkt.MH.OPCODE ==3
Yes
var.L2MC_INDEX = {DST_MODID[3:0], DST_PORT[4:0]}
tMCAST_TABLE[var.L2MC_INDEX].MCBITMAP
M
No
IP Multicast
Is Pkt.MH.OPCODE ==4
Yes
var.IMPC_INDEX={DST_MODID[3:0], DST_PORT[4:0]}
var.PORTBITMAP |= tMCAST_TABLE[var.IPMC_INDEX].IPMCPORTBITMAP
M
No
Unsupported MH Opcode
Drop the packet & Increment Ingess cRDISC Counter

Fig. 4

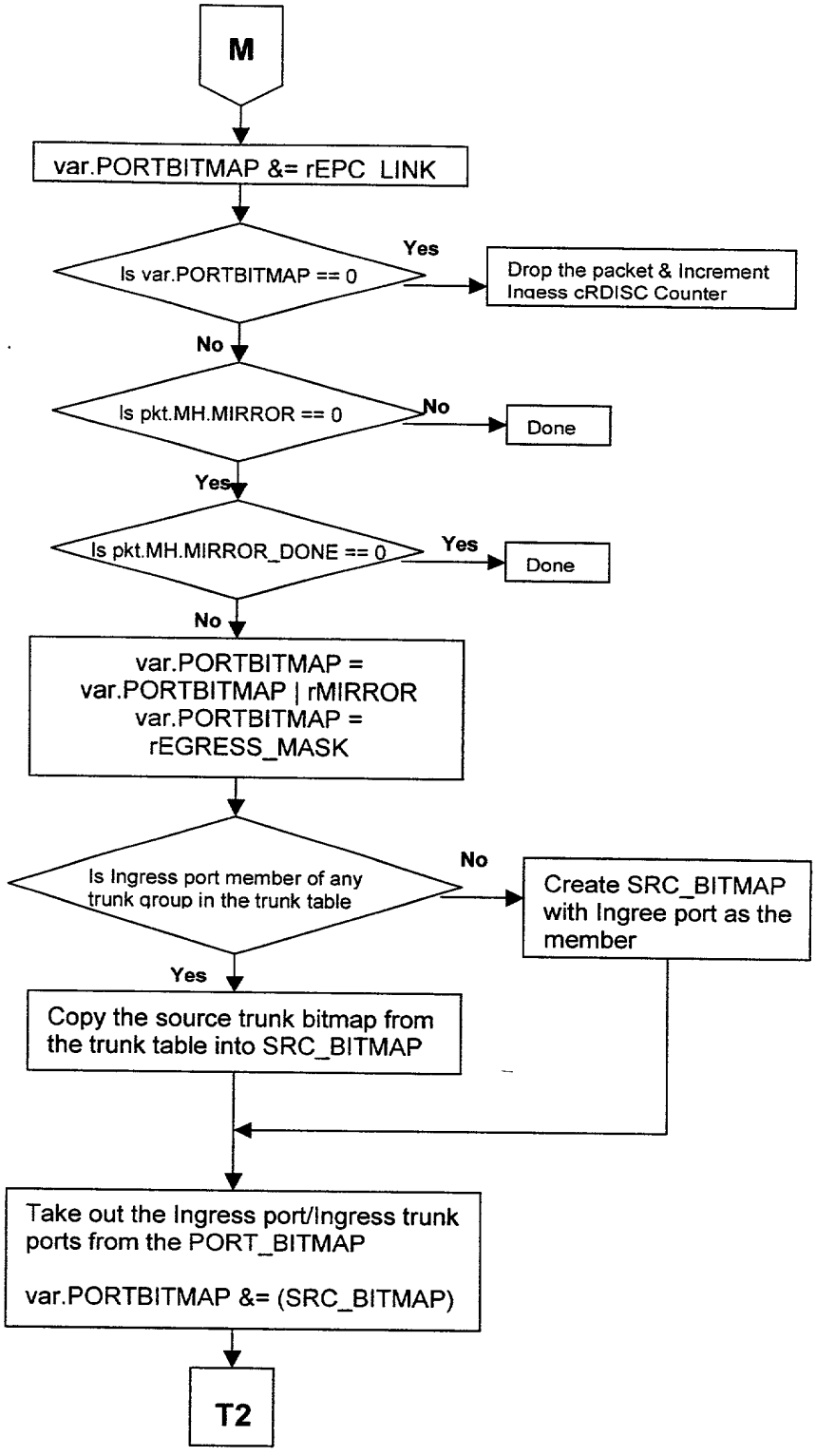
M
var.PORTBITMAP &= rEPC_LINK
Is var.PORTBITMAP == 0
Yes
Drop the packet & Increment Ingess cRDISC Counter
No
Is pkt.MH.MIRROR == 0
No
Done
Yes
Is pkt.MH.MIRROR_DONE == 0
Yes
Done
No
var.PORTBITMAP = var.PORTBITMAP | rMIRROR var.PORTBITMAP = rEGRESS_MASK
Is Ingress port member of any trunk group in the trunk table
No
Create SRC_BITMAP with Ingree port as the member
Yes
Copy the source trunk bitmap from the trunk table into SRC_BITMAP
Take out the Ingress port/Ingress trunk ports from the PORT_BITMAP
var.PORTBITMAP &= (SRC_BITMAP)
T2

Fig. 5

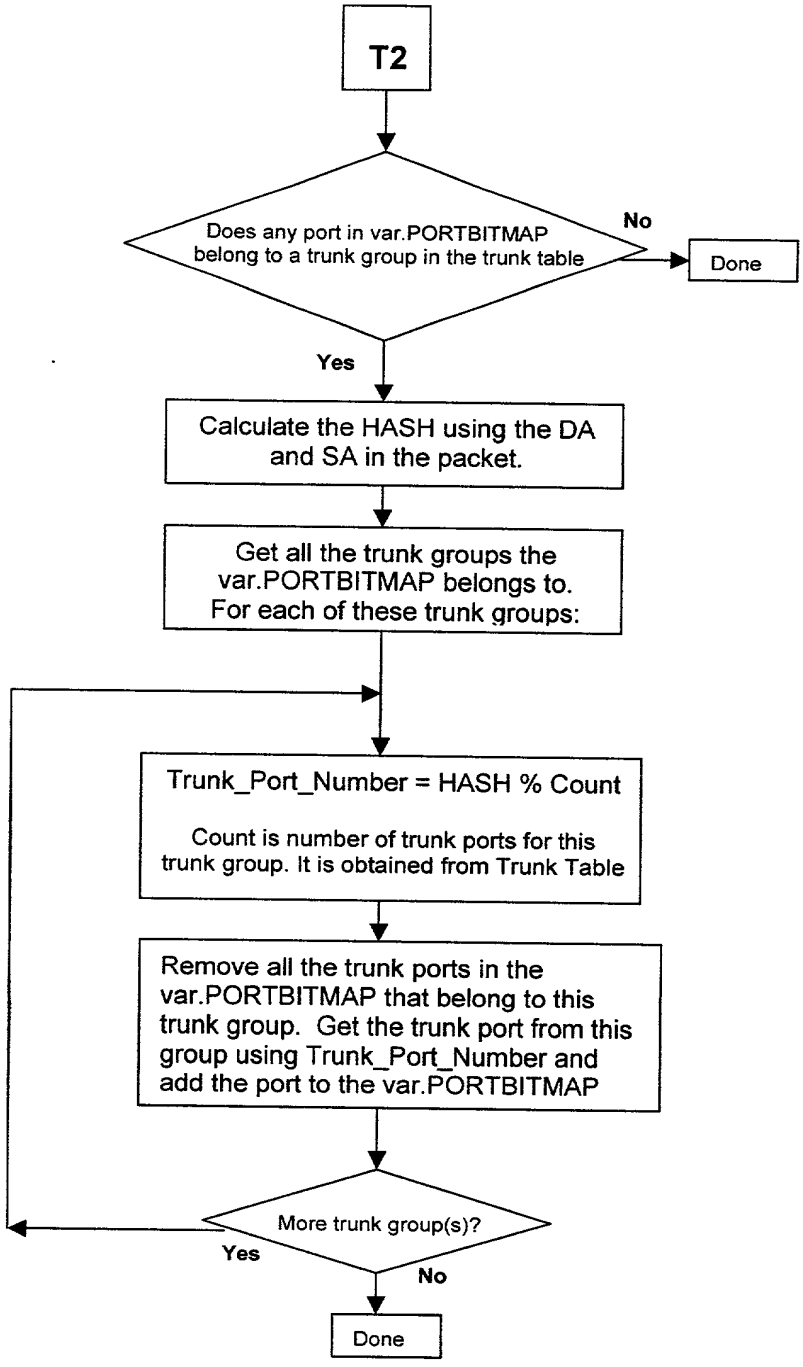
T2
Does any port in var.PORTBITMAP belong to a trunk group in the trunk table
No
Done
Yes
Calculate the HASH using the DA and SA in the packet.
Get all the trunk groups the var.PORTBITMAP belongs to. For each of these trunk groups:
Trunk_Port_Number = HASH % Count
Count is number of trunk ports for this trunk group. It is obtained from Trunk Table
Remove all the trunk ports in the var.PORTBITMAP that belong to this trunk group. Get the trunk port from this group using Trunk_Port_Number and add the port to the var.PORTBITMAP
More trunk group(s)?
Yes
No
Done

Fig. 6

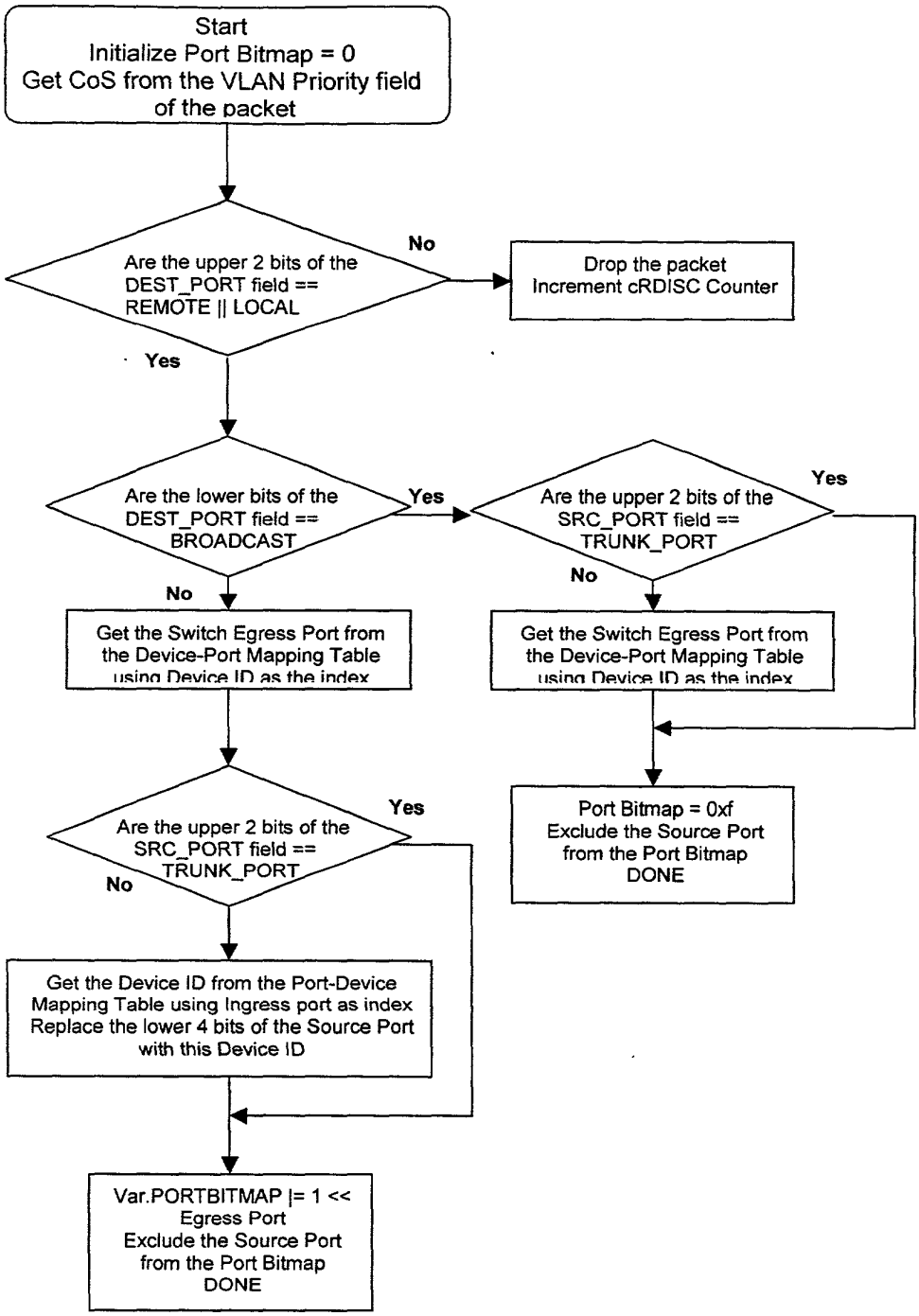
Start
Initialize Port Bitmap = 0
Get CoS from the VLAN Priority field of the packet
Are the upper 2 bits of the DEST_PORT field == REMOTE || LOCAL
No
Drop the packet
Increment cRDISC Counter
Yes
Are the lower bits of the DEST_PORT field == BROADCAST
Yes
Are the upper 2 bits of the SRC_PORT field == TRUNK_PORT
Yes
No
No
Get the Switch Egress Port from the Device-Port Mapping Table using Device ID as the index
Get the Switch Egress Port from the Device-Port Mapping Table using Device ID as the index
Are the upper 2 bits of the SRC_PORT field == TRUNK_PORT
Yes
No
Port Bitmap = 0xf
Exclude the Source Port from the Port Bitmap
DONE
Get the Device ID from the Port-Device Mapping Table using Ingress port as index
Replace the lower 4 bits of the Source Port with this Device ID
Var.PORTBITMAP |= 1 << Egress Port
Exclude the Source Port from the Port Bitmap
DONE

Fig. 7

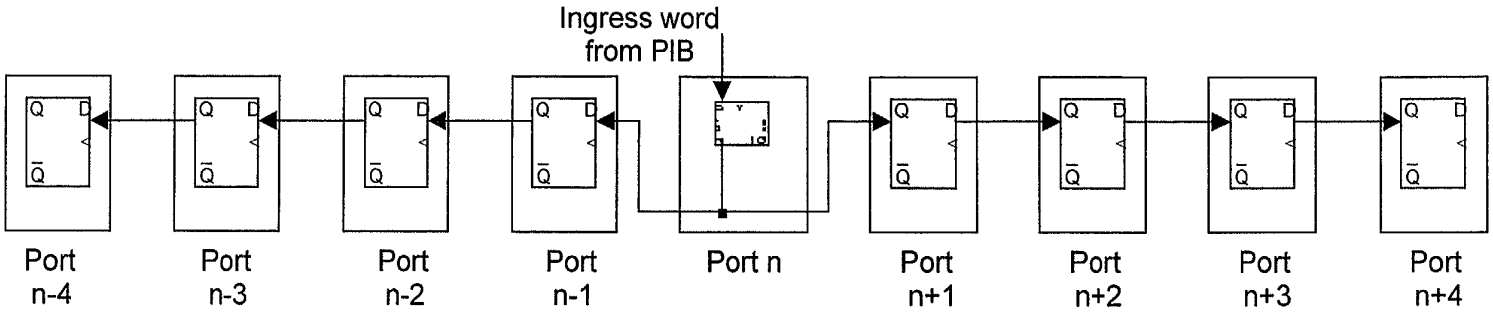
Ingress word
from PIB
Port
n-4
Port
n-3
Port
n-2
Port
n-1
Port n
Port
n+1
Port
n+2
Port
n+3
Port
n+4

Fig. 8

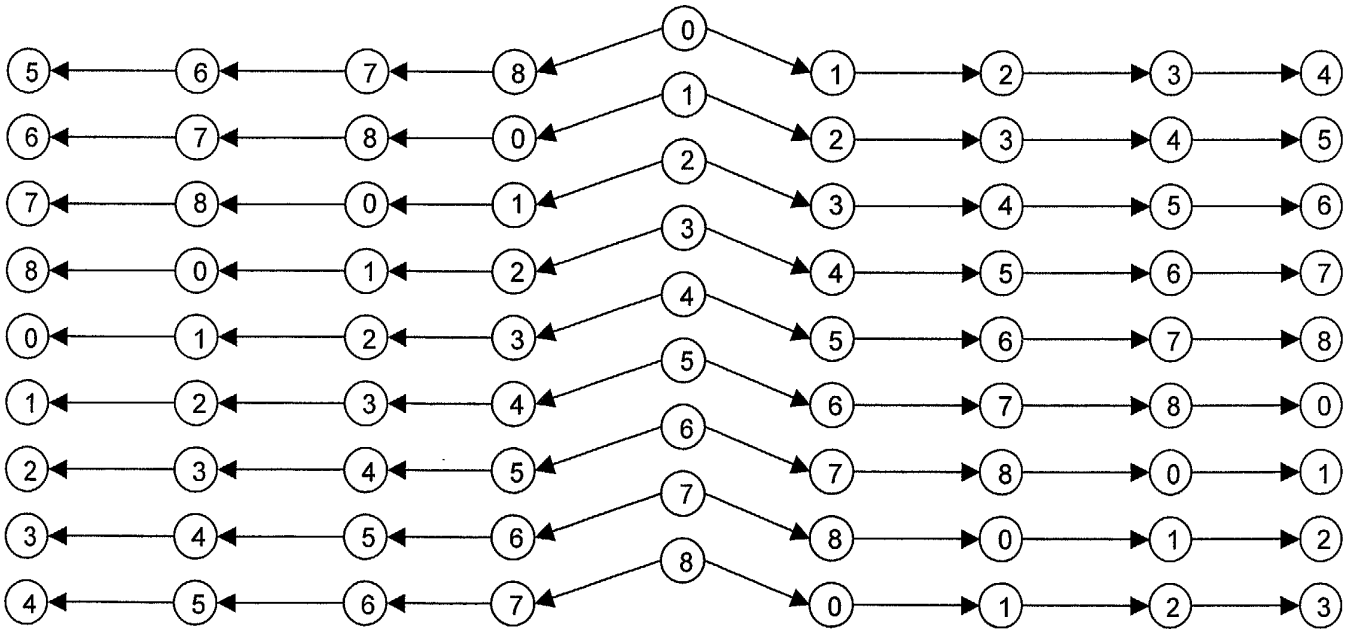

Fig. 9

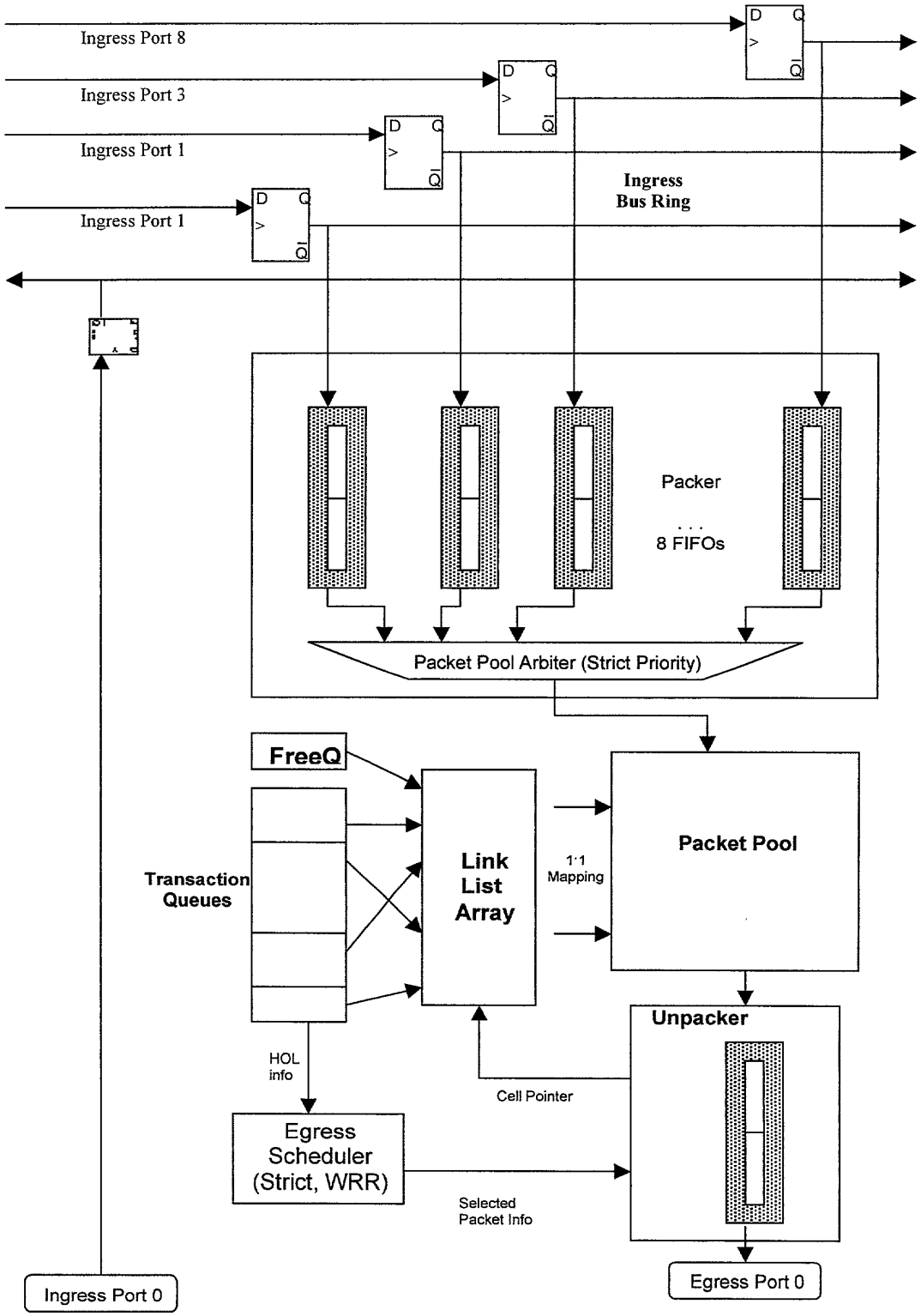
Ingress Port 8
Ingress Port 3
Ingress Port 1
Ingress Port 1
Ingress
Bus Ring
Packer
. . .
8 FIFOs
Packet Pool Arbiter (Strict Priority)
FreeQ
Transaction
Queues
Link
List
Array
1:1
Mapping
Packet Pool
Unpacker
HOL
info
Cell Pointer
Egress
Scheduler
(Strict, WRR)
Selected
Packet Info
Ingress Port 0
Egress Port 0

Fig. 10

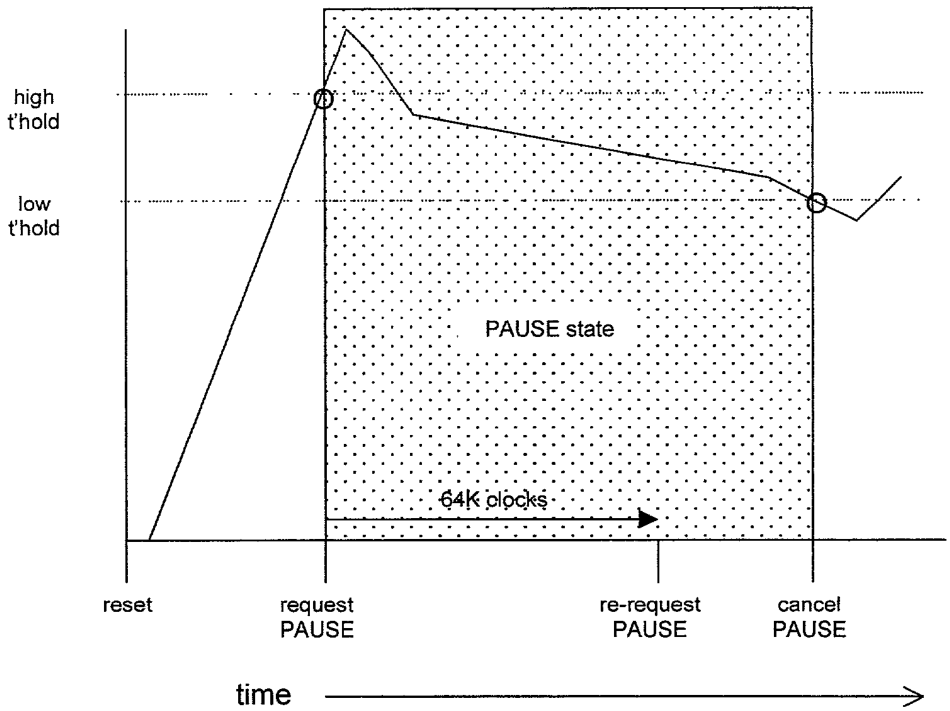

Fig. 11

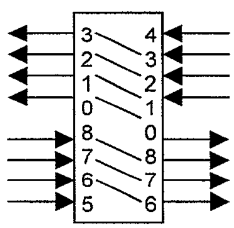

Fig. 12

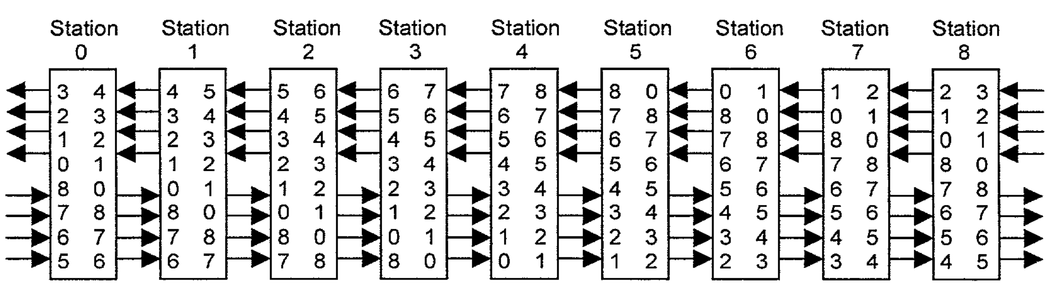

Fig. 13

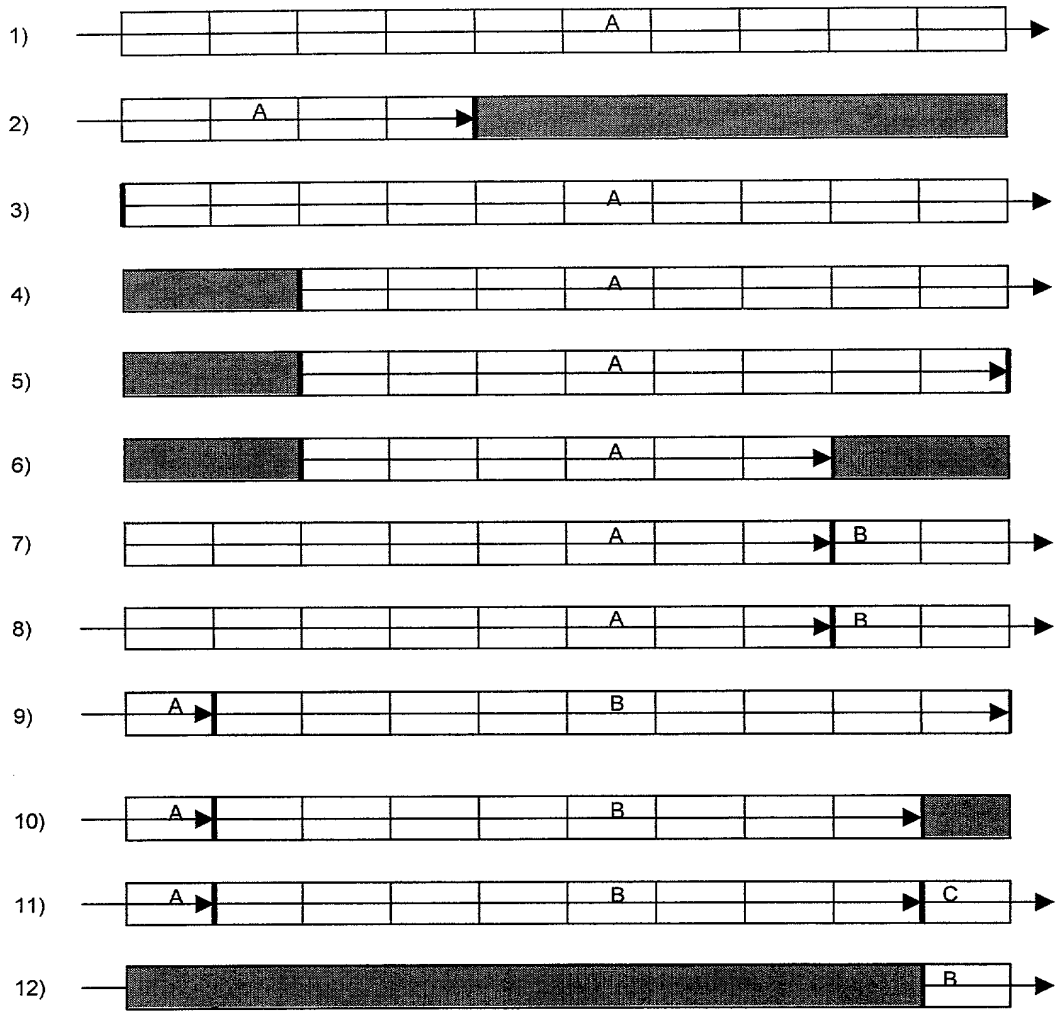
1)
A
2)
A
3)
A
4)
A
5)
A
6)
A
7)
A
B
8)
A
B
9)
A
B
10)
A
B
11)
A
B
C
12)
B

Fig. 14

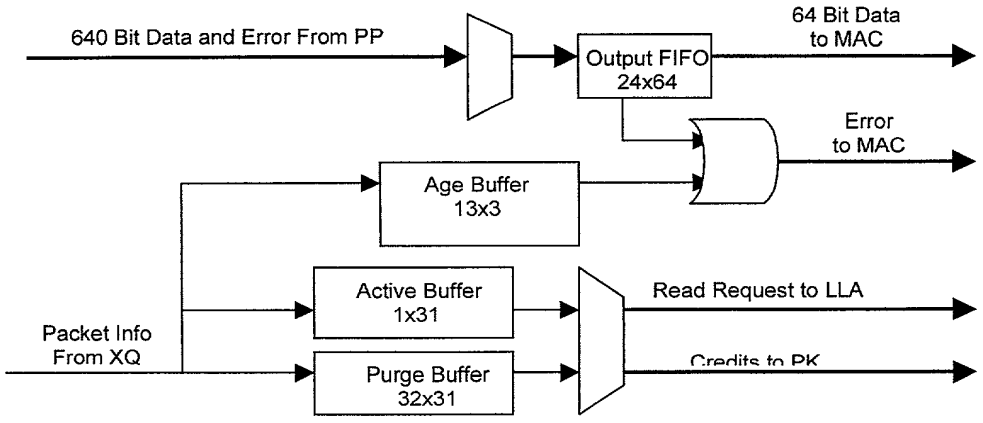
640 Bit Data and Error From PP
Output FIFO
24x64
64 Bit Data
to MAC
Error
to MAC
Age Buffer
13x3
Active Buffer
1x31
Read Request to LLA
Packet Info
From XQ
Purge Buffer
32x31
Credits to PK

Fig. 16

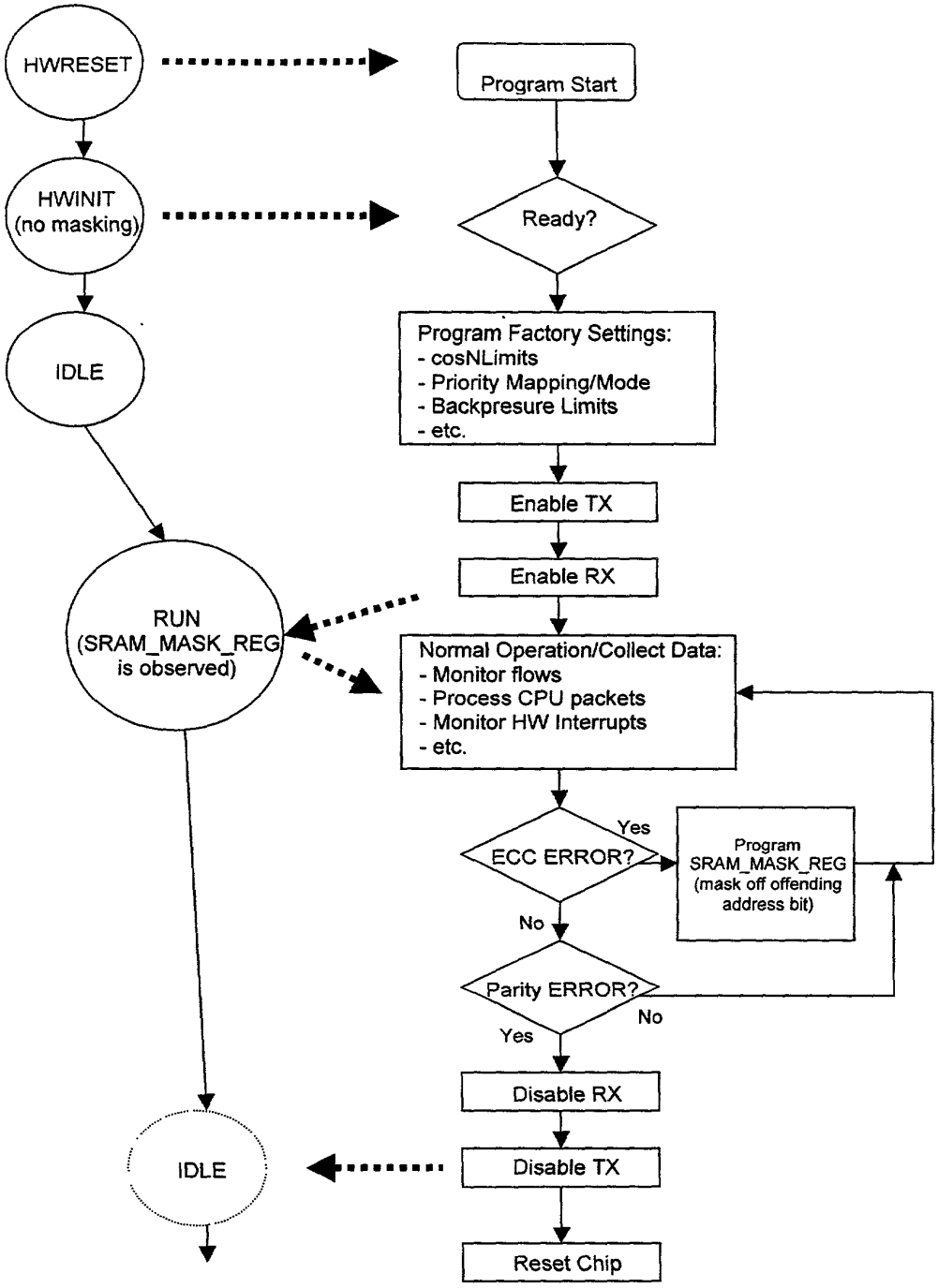
HWRESET
HWINIT
(no masking)
IDLE
RUN
(SRAM_MASK_REG
is observed)
IDLE
Program Start
Ready?
Program Factory Settings:
- cosNLimits
- Priority Mapping/Mode
- Backpresure Limits
- etc.
Enable TX
Enable RX
Normal Operation/Collect Data:
- Monitor flows
- Process CPU packets
- Monitor HW Interrupts
- etc.
ECC ERROR?
Yes
No
Program
SRAM_MASK_REG
(mask off offending
address bit)
Parity ERROR?
No
Yes
Disable RX
Disable TX
Reset Chip

Fig. 15